

---

---

# NIRSPEC

UCLA Astrophysics Program

U.C. Berkeley

W.M.Keck Observatory

---

---

Fred Lacayanga

September 14, 1998

## NIRSPEC Software Programming Note 29.00 Adding Keywords

- Add keyword information into the keyword table in `nirspec.h` in the server code directory `/kroot/kss/nirspec/keyword`.

```
/*
 * Keyword table structure
 */
typedef struct {
    char      keyword[80];          /* NIRSPEC keyword name      */
    KTL_DATATYPE datatype;        /* keyword data type         */
    char      initval[80];         /* initial value             */
    double    minval;              /* minimum value             */
    double    maxval;              /* maximum value             */
    int       cli;                  /* valid CLI Tcl command if TRUE */
    int       argnum;              /* # CLI command args; 0 = any */
    int       cid;                  /* transputer command id     */
    int       broadcast;           /* broadcast flag (TRUE or FALSE)*/
    int       rwflag;              /* readable/writable flag     */
    int       access;              /* keyword classification     */
    char      units[80];           /* Units field                */
    char      help[80];            /* single line help          */
    char      alias[80];           /* command alias              */
} KEYWORD_TABLE;
```

For example:

```
static KEYWORD_TABLE KeywordTable[] = {
    "irotstep", KTL_INT,      "0",          -108000,108000,
                TRUE,  1, CID_MOT_IROT_STEP,
                TRUE,  WRABLE,      1, "steps",
                "move image rotator in steps", "",
}
}
```

- Add server code for the keyword in the `rpc_write_1( )` function in `nrpc_svc_proc.c`.

```
/* Step */
else if ( strcasecmp( arg->keyword, "irotstep" ) == 0 ) {
    broadcast_status("irotstat", "SLEW");
    if ( ( status = STP_moveSteps( MOT_IROT, param ) ) < 0 ) {
        broadcast_status("irotstat", "UNKNOWN");
        broadcast_status("tspstat", "TSP SEND ERROR");
        res.status = status;
        res.errmsg = STPErrMsg[-res.status-STP_ERR_CODE0];
        return &res;
    }
}
```

- If the keyword sends a transputer command that just needs to be passed down the link without any interpretation (i.e. `sample.spec`, `getcryotemp`), then no server code needs to be written in `rpc_write_1( )` because at the very end of the function, the cid and parameter is sent down to the transputers:

```

/*
 * Finally, send any other commands straight to the TSP link
 */
else
    res.status = TSPCom_sendCommand( cid, param );

```

- If the keyword sends a request to the transputers, then add server code to interpret the messages sent back from the transputers in the `tsplink_msg_process( )` function of `nrpc_server.c`.

```

/* stepping */
else if ( cid == CID_MOT_IROT_STEP ) {

    if ( param == 1 ) {
        if ( MotorInitStatus[MOT_IROT] == TRUE )
            broadcast_tsp_msg("irotstat", "OK", 0, 0);
        else
            broadcast_tsp_msg("irotstat", "UNKNOWN", 0, 0);
        MotorMove[ MOT_IROT ] = FALSE;
        STP_checkRange( MOT_IROT );
    }

    STP_moveErrorInfo( MOT_IROT, param );
}

```

- If the keyword is broadcast and the client needs to act upon this broadcast, then you must create interest in this keyword in `create_interest.c` in the client code directory `/kroot/kss/nirspec/ui/xnirspec`.

```
EXPRESS_INTEREST( "tspstat",  tspstat_callback );
```

And add a callback function in `callbacks.c` in the client code directory.

```

void tspstat_callback( keyword, user_data, call_data, context )
char *keyword;          /* keyword */
void *user_data;       /* unused */
KTL_POLYMORPH *call_data; /* contains new value */
KTL_CONTEXT *context;  /* command context (unused) */
{
    if ( (strcmp( call_data->s, "IROT motor move success.") != 0) ||
          (strcmp( call_data->s, "IROT motor is busy.") != 0) ) {
        strcpy( VdpBuf_status_message, call_data->s );
        DV_updateScreen( SCREEN_INST );
    }
}

```